Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended): A method for conducting an exchange of data with a terminal-based application program comprising:

(a) mapping a plurality of available states within a terminal data stream of the terminal-based application program to respective discrete state definitions within a finite state machine, including:

(1) interpreting any element, terminal command, data item, or sequence of terminal commands and data items within the terminal data stream as a discrete state having a respective one of the state definitions, and

(2) using an object model containing a set of interfaces, said interfaces being utilized as a basis for the state definitions;

(b) prompting a user for rules criteria and expected outcomes of the respective state definitions;

(c) aggregating the plurality of available states within the terminal data stream to eliminate redundant states, including:

(1) accessing data in the plurality of available states;

(2) unifying and resolving the data in the plurality of available states to eliminate the redundant states; and,

(d) defining a plurality of state transition rules which are utilized to manipulate the state definitions within the finite state machine.

2. (Original): The method of claim 1, wherein the object model contains the set of interfaces and a set of classes.

3. (Original): The method of claim 1, wherein manipulation of the state transition rules allows for two-way communication between a client and the terminal-based application program.

4. (Original): The method of claim 1, further comprising creating one or more data sets, each comprising a definable set of data elements from within the terminal-based application program.

5. (Original): The method of claim 1, wherein the object model is a distributed object transaction model, which allows for access to data from the terminal-based application program by any local or remote client service, system, or application.

6. (Previously Presented): The method of claim 1 further comprising unifying and resolving multiple terminal-based applications through unification and resolution of a plurality of instances of a program that performs steps (a) and (d).

7. (Previously Presented): The method of claim 6, further comprising creating a plurality of data sets, each data set being formed through a unification of multiple data sets from the plurality of instances of the program that performs steps (a) and (d).

8. (Original): The method of claim 6, wherein each object model is a distributed object transaction model which allows for access to data from the terminal-based application program by any local or remote client service, system, or application.

9. (Previously Presented): The method of claim 6, further comprising using an expert system which accesses and resolves data items from the plurality of instances of the program that performs steps (a) and (d) and translates them into cohesive super sets of data.

10. (Previously Presented): The method of claim 9, further comprising using the object model for creating data resolution and translation rules.

11. (Previously Presented): The method of claim 9 further comprising using the object model for definition of actions to proactively resolve data errors or discrepancies across an underlying plurality of instances of the program that performs steps (a) and (d).

12. (Previously Presented): The method of claim 1, further comprising altering an interface presented to a user of the terminal-based application program through addition of one of a group consisting of new screens and new data fields within existing screens, wherein the new screens and new data fields are populated with data retrieved from an alternate data source.

13. (Original): The method of claim 12, further comprising monitoring the terminal data stream.

14. (Original): The method of claim 13, further comprising:

recognizing pre-defined states within the terminal data stream, which define one or more states during a user's interaction with the terminal-based application program; and

presenting the new data screens or fields to the user.

15. (Previously Presented): The method of claim 12, wherein the object model describes interaction between the alternate data source and a program that performs steps (a) and (d).

16. (Original): The method of claim 12, wherein the object model describes:

the addition of new user screens or the addition of new data fields to existing application screens, and

interaction between the user and the new screens or fields.

17. (Original): The method of claim 1 further comprising using software tools to automate creation and maintenance of an integration system based on knowledge of a domain of the terminal-based application program.

18. (Previously Presented): A system for conducting an exchange of data with a terminal-based application program comprising:

a finite state machine, in which a plurality of available states within a terminal data stream of the terminal-based application program are mapped to respective discrete state definitions, including:

means for interpreting any element, terminal command, data item, or sequence of terminal commands and data items within the terminal data stream as a discrete state having a respective one of the state definitions;

means for using an object model containing a set of interfaces, said interfaces being utilized as a basis for the state definitions;

means for prompting a user for rules criteria and expected outcomes of the respective means for aggregating the plurality of available states within the terminal data stream to eliminate redundant states;

and

means for defining a plurality of state transition rules which are utilized to manipulate the state definitions within the finite state machine.

19. (Original): The system of claim 18, wherein the object model contains the set of interfaces and a set of classes.

20. (Canceled)

21. (Canceled)

22. (Canceled)

23. (Original): The system of claim 18 wherein multiple terminal-based applications are unified and resolved through unification and resolution of a plurality of instances of the finite state machine.

24. (Original): The system of claim 23, further comprising a plurality of data sets, each data set being formed through a unification of multiple data sets from the plurality of instances of the finite state machine.

25. (Original): The system of claim 23, wherein each object model is a distributed object transaction model which allows for access to data from the terminal-based application program by any local or remote client service, system, or application.

26. (Original): The system of claim 23, further comprising an expert system which accesses and resolves data items from the plurality of instances of the finite state machine and translates them into cohesive super sets of data.

27. (Previously Presented): The system of claim 26, wherein the object model is used for creating data resolution and translation rules.

28. (Previously Presented): The system of claim 26 wherein the object model is used for definition of actions to proactively resolve data errors or discrepancies across an underlying plurality of instances of the finite state machine.

29. (Previously Presented): The system of claim 18, further comprising an interface presented to a user of the terminal-based application program, the interface formed through addition of one of a group consisting of new screens and new data fields within existing screens, wherein the new screens and new data fields are populated with data retrieved from an alternate data source.

30. (Original): The system of claim 29, further comprising means for monitoring the terminal data stream.

31. (Original): The system of claim 30, wherein:

pre-defined states are recognized within the terminal data stream, which define one or more states during a user's interaction with the terminal-based application program; and

the new data screens or fields are presented to the user.

32. (Original): The system of claim 29, wherein the object model describes interaction between the alternate data source and the finite state machine.

33. (Original): The system of claim 29, wherein the object model describes:

the addition of new user screens or the addition of new data fields to existing application screens, and

interaction between the user and the new screens or fields.

34. (Original): The system of claim 18 further comprising software tools that automate creation and maintenance of an integration system based on knowledge of a domain of the terminal-based application program.

35. (Previously Presented): The system of claim 18, further comprising a tool which automates capture of a terminal data stream and the creation of state definitions for a particular terminal-based application.

36. (Original): The system of claim 35, wherein the tool allows the user to define the data sets which will be made available.

37. (Original): The system of claim 36, wherein the tool allows the user to define state transition rules to access and manipulate the data sets, to read and write data elements, using a point-and-click flowchart-style interface.

38. (Original): The system of claim 35, further comprising software tools which automate creation and maintenance of a unification and resolution system based on a knowledge of underlying integration systems being unified.

39. (Original): The system of claim 35, wherein the tool allows the user to define the data super-sets which will be made available by the system.

40. (Original): The system of claim 35, wherein the tool allows the user to create and define data unification and resolution rules.

41. (Previously Presented): The system of claim 35, wherein the tool allows the user to define rules to manage data errors and discrepancies in the terminal data stream.

42. (Previously Presented): A computer readable medium encoded with computer program code, wherein when the computer program code is executed by a processor, the processor performs a method for conducting an exchange of data with a terminal-based application program comprising:

(a) mapping a plurality of available states within a terminal data stream of the terminal-based application program to respective discrete state definitions within a finite state machine, including:

(1) interpreting any element, terminal command, data item, or sequence of terminal commands and data items within the terminal data stream as a discrete state having a respective one of the state definitions, and

(2) using an object model containing a set of interfaces, said interfaces being utilized as the basis for the state definitions;

(b) prompting a user for rules criteria and expected outcomes of the respective state definitions;

(c) aggregating the plurality of available states within the terminal data stream to eliminate redundant states; and,

(d) defining a plurality of state transition rules which are utilized to manipulate the state definitions within the finite state machine.

43. (Previously Presented): The computer readable medium of claim 41, wherein the method further comprises unifying and resolving multiple terminal-based applications through unification and resolution of a plurality of instances of a program that performs steps (a) and (d).

44. (Previously Presented): The computer readable medium of claim 41, wherein the method further comprises using software tools to automate creation and maintenance of an integration system based on knowledge of a domain of the terminal-based application program.

45. (Previously Presented) A method of defining and configuring the exchange of data with a terminal-based application, comprising:

(a) accessing a targeted legacy application on at least one legacy application server;

(b) evaluating operated information and defined rules on the targeted legacy application;

(c) modeling the targeted legacy application with a legacy application profile, screen, and data stream definitions;

(d) providing automated and dynamically directed execution and runtime processing environment operating concurrently and coordinated across the at least one legacy application server;

(e) providing processed legacy information objects using the objectification definitions in customizable formats and structures for access by multiple requesting applications.

46. (New) The method of claim 14, wherein the unifying and resolving of the data in the plurality of available states to eliminate the redundant states further comprises:

synchronizing the data in the plurality of available states; and

correcting spelling errors or data entry errors in the data in the plurality of available states.